Volodymyr Afanasiev Follow

Sep 17, 2018 · 3 min read

Список из 16-ти artisan:make команд с параметрами

Laravel имеет удивительный набор команд artisan, вероятно, наиболее часто используются **make:xxx** —такие как, **make:model** или make:migration и т. д. Но знаете ли вы все 16 из них? И, более того, знаете ли вы их параметры, которые могут помочь сделать код еще быстрее?

Во-первых, есть команда **php artisan** список, который дает нам все команды, как это:

```
make:auth              Основные представления и маршруты входа и регистрации
в скаффолде
  make:command           Создание новой команды Artisan
  make:controller        Создание нового класса контроллера
  make:event             Создание нового класса события
  make:job               создание нового класса заданий
  make:listener          создание нового класса прослушивателя событий
  make:mail              Создание нового класса электронной почты
  make:middleware        Создание нового класса посредника
  make:migration         Создание нового файла миграции
  make:model             Создание нового класса модели Eloquent
  make:notification      Создание нового класса уведомлений
  make:policy            Создание нового класса политики
  make:provider          Создание нового класса сервис провайдера
  make:request           Создание нового класса запроса формы
  make:seeder            Создание нового seeder класса
  make:test              Создание нового тестового класса
```

Но это не дает нам никакой информации о параметрах или параметрах этих команд. Так что я хочу сделать обзор каждого из них, начиная с наиболее часто используемых.

Для этого мы погрузимся в реальный код фреймворка, внутри папки /vendor/laravel/framework/src/Illuminate, и проверим, какие опции и недокументированные возможности у нас есть для каждой команды.

1. Make:controller

Эта команда создает новый файл контроллера в папке app/Http/Controllers.

Пример использования:

```
php artisan make:controller UserController
```

Параметры:

```
--resource
```

Контроллер будет содержать метод для каждой операции, ресурса — index(), create(), store(), show(), edit(), update(), destroy().

```
--model=Photo
```

Если вы используете маршрут привязка модели и методы контроллера ресурса типа намек экземпляр модели.

```
--parent=Photo
```

Официально недокументированный параметр, в коде написано “Generate a nested resource controller class”(Создание класса контроллера вложенных ресурсов), но для меня он не смог правильно сгенерировать контроллер. Так что, вероятно, работа продолжается.

2. Make:model

Создание нового класса модели Eloquent.

Пример использования:

```
php artisan make:model Photo
```

Параметры:

```
--migration
```

Создаст новый файл миграции для модели.

```
--controller
```

Создаст новый контроллер для модели.

```
--resource
```

Указывает, должен ли создаваемый контроллер быть контроллером ресурсов.

Да, у вас все правильно, вы можете сделать это так:

```
php artisan make:model Project --migration --controller --resource
```

Или даже короче:

```
php artisan make:model Project -mcr
```

3. Make:migration

Создает новый файл миграции.

Пример использования:

```
php artisan make:migration create_projects_table
```

Параметры:

```
--create=Table
```

Создаваемой таблицы.

```
--table=Table
```

Таблица для переноса.

```
--path=Path
```

Расположение, в котором должен быть создан файл миграции.

4. make:seeder

Создайте новый seeder класс базы данных.

Пример использования:

```
php artisan make:seeder BooksTableSeeder
```

Параметры: нет.

5. make:request

Создайте новый класс запроса формы в папке **app/Http/Requests**.

Пример использования:

```
php artisan make:request StoreBlogPost
```

Параметры: нет.

6. make:middleware

Создает новый класс посредника.

Пример использования:

```
php artisan make:middleware CheckAge
```

Параметры: нет.

7. make:policy

Создает новый класс политики.

Пример использования:

```
php artisan make:policy PostPolicy
```

Параметры:

Модель, к которой применяется политика.

8. make:auth

Пример использования:

```
php artisan make:auth
```

Авторизация и регистрация просмотров и маршрутов.

Параметры:

```
--views
```

Только представления проверки подлинности.

```
--force
```

Перезаписать существующие представления по умолчанию.

9. make:command

Создайте новую команду Artisan.

Пример использования:

```
php artisan make:command SendEmails
```

Параметры:


```
--command=Command
```

Команда терминала, которая должна быть назначена.

10. make:event

Создайте новый класс событий.

Пример использования:

```
php artisan make:event OrderShipped
```

Параметры: нет.

11. make:job

Создайте новый класс задания.

Пример использования:

```
php artisan make:job SendReminderEmail
```

Параметры:

```
--sync
```

Указывает, что задание должно быть синхронным.

12. make:listener

Создает новый класс прослушивателя событий.

Пример использования:

```
php artisan make:listener SendShipmentNotification
```

Параметры:

```
--event=Event
```

Класс событий прослушивается.

```
--queued
```

Указывает, что прослушиватель событий должен быть поставлен в очередь.

13. make:mail

Создает новый класс электронной почты.

Пример использования:

```
php artisan make:mail OrderShipped
```

Параметры:

```
--markdown
```

Создайте новый шаблон Markdown для отправляемой почты.

14. make:notification

Создает новый класс уведомлений.

Пример использования:

```
php artisan make:notification InvoicePaid
```

Параметры:

```
--markdown
```

Создает новый шаблон Markdown для уведомления.

15. make:provider

Создает новый класс поставщика услуг.

Пример использования:

```
php artisan make:provider RiakServiceProvider
```

Параметры: нет.

16. make:test

Создает новый тестовый класс.

Пример использования:

```
php artisan make:test UserTest
```

Параметры:

```
--unit
```

Создает модуль(или другие фичи) теста.

Итак, вот они — 16 команд. Что-нибудь пропало? Или вы хотели бы увидеть некоторые новые параметры для ваших любимых команд? Поделитесь в комментариях!

P. S. Мы недавно выпустили пакет, который добавляет новую команду — make:api. Смотрите на API Generator on GitHub.

Перевод с https://quickadminpanel.com/blog/list-of-16-artisan-make-commands-with-parameters/

Также рекомендую ознакомиться с:

Запуск команд PHP Artisan на Shared Hosting серверах

Сокращенные консольные команды в Laravel

Laravel Laravel Framework Laravel Artisan Laravel 5 Cheat Sheet PHP